Merging Expressive Spatial Ontologies Using Formal Concept Analysis with Uncertainty Considerations

Olivier Curé

Chapter 1
Merging Expressive Spatial Ontologies using Formal Concept Analysis with Uncertainty Considerations

Olivier Curé

Abstract In this chapter, we present a solution to the problem of merging structures that represent the conceptual layer of some information systems. The kind of structures we are studying correspond to expressive ontologies formalized in Description Logics. The proposed approach creates a merged ontology which captures the knowledge of a set of source ontologies. A main constraint to our solution consists in the fact that instances associated to the concepts of the source ontologies are available. Then it is possible to apply the techniques associated to Formal Concept Analysis. The main contributions of this work are (i) enabling the creation of concepts not originally in the source ontologies, (ii) providing a definition to these concepts in terms of elements of the source ontologies and (iii) handling the creation of merged ontologies based on the uncertainties encountered at the object and alignment levels. This approach is particularly useful in domains where ontologies are intensively exploited. This is typically the case for spatial information where for instance, the nature of land parcels can be characterized by a geographical ontology.

1.1 Introduction

The information stored in Geographical Information Systems (GIS) usually needs to be exchanged and integrated between multiple applications. These tasks raise several important issues due to format and semantics heterogeneity but also to handle several forms of uncertainty that are encountered. For instance, we can distinguish between uncertainties at the application domain level and uncertainties at the integration/exchange level. Concerning the first

Université Paris-Est, IGM Terre Digitale,
Marne-la-Vallée, France
e-mail: ocure@univ-mlv.fr

type of uncertainty, the information stored into GIS are almost always sampled. Some sampling uncertainties are related to geodesy's positional accuracy or semantical accuracy when characterizing the nature of a sample. Uncertainty occurring in matching and mapping operations are considered as an important issue and several solutions have already been proposed, see [16] for a survey.

In this chapter, we are also interested in the semantic issues. The integration of ontologies within the information system, usually to represent its conceptual layer, has been one approach to respond to these concerns. This is the case in GIS and the spatial information domain in general where several ontologies emerged recently. For instance, the Semantic Web for Earth and Environmental Terminology (SWEET) [29] provides an upper-level ontology for Earth system science. In the context of space parcels, the CORINE land cover [15] and the ATKIS catalogue [1] are terminologies that enable to characterize land-use types. Application designers also frequently generate their own ontologies to reply to some special needs. These ontologies may be created from scratch or by an alignment and extension from existing ontologies. Hence, it is well-known that many ontologies coexist in some specific domains, e.g. geographical information and medicine.

With so many ontologies being produced, it is inevitable that some of their content overlaps and possibly disagrees on some concepts. In order to support ontology interoperability, it is required that these ontologies can be semantically related. Thus ontology mediation [14] becomes a main concern. Ontology mediation enables to share data between heterogeneous knowledge bases, and allows applications to reuse data from different knowledge bases. Ontology mediation takes two distinguished forms: (i) ontology mapping, where the correspondences between elements of two ontologies are stored separately from the ontologies. The correspondences are generally represented using axioms formulated in a peculiar mapping language. (ii) ontology merging, which consists in creating a new ontology from the union of source ontologies. The merged ontology is supposed to capture all the knowledge of the sources.

Ontology mediation is an active research field where many kinds of solutions have been proposed: schema-based, instance-based, machine learning-inspired, hybrid approaches; see [20], [16] for surveys on this domain. Thus the methods used in ontology mediation usually depend on the kind of information one can access about the local ontologies. For instance, the availability of instance datasets are highly desirable and generally ensure good mediation results. But the efficiency of these methods also depends on the kind of source ontologies the system is dealing with. In [23], the author presents an ontology spectrum which characterizes the expressiveness of several ontology solutions.

In this paper, we are interested in declarative and logic-based formalisms to represent ontologies. In fact, we consider one of the currently most popular formalism, i.e. Description Logics (DLs). Popular because DLs are underpinning the Web Ontology Language (OWL) [7] proposed by the World Wide

Web Consortium. Thus this language is being used to represent a large number of ontologies in domains as diverse as social networking, medicine, bioinformatics and spatial information. Part of this popularity is also due to the availability of a number of ontology tools such as editors, e.g. Protégé, Swoop, and reasoners, e.g. Pellet, Racer, Fact, HermiT, to infer, usually with sound and complete methods, implicit knowledge from the explicitly represented one.

In this chapter, we propose a solution to the ontology merging problem which is based on the techniques of Formal Concept Analysis (FCA) [17]. It extends [8] by dealing with expressive ontologies and their concept descriptions. FCA algorithms are machine learning techniques that enable the creation of a common structure, which may reveal some associations between elements of the original structures. Thus it requires that some elements from the source ontologies can be attached to a same observable item. Starting from this assumption, the processing of our FCA-based algorithms provides a merged ontology.

Our solution extends existing FCA-based systems for ontology merging in the following way: (i) we provide a method to create concepts not originally in the source ontologies, (ii) we define emerging concepts in terms of elements of the source ontologies and (iii) we handle the creation of merged ontologies based on the uncertainty underlying the extension and alignment of source concepts. Step (i) is the classical approach named *ontology alignment* in FCA literature. Steps (ii) and (iii) are an extension of this alignment and exploit concept descriptions, DL reasoner functionalities and notions from possibility theory.

The paper is organized as follows: in Section 1.2, we present some basic notions about FCA, the $\mathcal{ALC}$ description logic and possibilistic logic. In Section 1.3, we detail our method which enables to create an expressive merged ontology. The main steps are: concept generation, axiomatization of emerging concepts and optimization of the resulting ontology. Section 1.4 proposes a solution to deal with the different forms of uncertainty encountered. Section 1.5 relates our work with existing systems in ontology merging and collaborations between FCA methods and DLs. Section 1.6 concludes this chapter.

1.2 Background

1.2.1 Formal Concept Analysis and Galois connection

FCA is the process of abstracting conceptual descriptions from a set of objects described by attributes [17]. We use some of the methods associated to FCA to merge geographical ontologies. Intuitively, this means that we merge several ontologies in a context consisting of a set of objects (the extent),

a set of attributes (the intent), one for each ontology, and a set of correspondences between objects and attributes. FCA is based on the notion of a *formal context*.

Definition 1: A formal context is a triple $\mathcal{K} = (G, M, I)$, where G is a set of objects, M is a set of attributes and I is a binary relation between G and M, i.e. $I \subseteq G \times M$. For an object g and an attribute m, $(g, m) \in I$ is read as "object g has attribute m".

Given a formal context, we can define the notion of formal concepts:

Definition 2: For $A \subseteq G$, we define $A' = \{m \in M | \forall g \in A : (g, m) \in I\}$ and for $B \subseteq M$, we define $B' = \{g \in G | \forall m \in B : (g, m) \in I\}$. A formal concept of $\mathcal{K}$ is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$.

The hierarchy of formal concepts is formalized by $(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2$ and $B_2 \subseteq B_1$. The concept lattice of $\mathcal{K}$ is the set of all its formal concepts with the partial order $\leq$. This hierarchy of formal concepts obeys the mathematical axioms defining a lattice, and is called a concept lattice (or Galois lattice) since the relation between the sets of objects and attributes is a Galois connection.

We now introduce the notion of Galois connection which is related to the idea of order and plays an important role in lattice theory, universal algebras and recently in computer science [5]. Let $(P, \preceq)$ and $(Q, \preceq)$ be two partially ordered sets (poset). A Galois connection between P and Q is a pair of mappings (Φ, Ψ) such that $\Phi : P \to Q$, $\Psi : Q \to P$ and:

- $x \preceq x'$ implies $\Phi(x) \succeq \Phi(x')$,
- $y \preceq y'$ implies $\Psi(y) \succeq \Psi(y')$,
- $x \preceq \Psi(\Phi(x))$ and $y \preceq \Psi(\Phi(y))$

for x,x' $\in$ P and y,y' $\in$ Q.

Several algorithms have been proposed to compute a concept lattice, some optimized ones are proposed in [6]. Intuitively, such an algorithm starts with the complete lattice of the power set of all individuals (the extent), respectively for attributes (the intent) and retains only the nodes closed under the connection. That is beginning with a set of attributes, the algorithm determines the corresponding set of objects which itself provides an associated set of attributes. If this set is the initial one, then it is closed and preserved otherwise the node is removed from the lattice.

1.2.2 The Description Logic $\mathcal{ALC}$

DLs are a family of knowledge representation formalisms allowing to reason over domain knowledge, in a formal and well-understood way. Central DL notions are concepts (unary predicates), roles (binary predicates) and individuals. A concept represents a set of individuals while a role determines

a binary relationship between concepts. DLs are a fragment of first-order logic and thus concepts and roles are designed according to a syntax and a semantics. Some of the main assets of this family of formalisms are decidability, efficient reasoning algorithms and the ability to propose a hierarchy of languages with various expressive power.

A key notion in DLs is the separation of the terminological (or intensional) knowledge, called a TBox, to the assertional (or extensional) knowledge, called the ABox. The TBox is generally considered to be the ontology. Together, a TBox and an ABox represent a Knowledge Base (KB), denoted $KB = \langle TBox, ABox \rangle$.

The TBox is composed of "primitive concepts" which are ground descriptions that are used to form more complex descriptions, "defined concepts" which are designed using a set of constructors of the description language, e.g. conjunction($\sqcap$), disjunction ($\sqcup$), negation ($\neg$), universal ($\forall$) and existential ($\exists$) value quantifiers, etc.

The description language we are using in this paper correspond to $\mathcal{ALC}$ (Attributive Language with Complements). Concept descriptions in this language are formed according to the following syntax rule, where the letter A is used for atomic concepts, the letter R for atomic roles and the letters C and D for concept descriptions:

$C, D ::= \bot \mid \top \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$

The terminological axioms accepted by $\mathcal{ALC}$ are sentences of the form $C \sqsubseteq D$ and which are called General Concept Inclusion (GCI).

The semantics generally adopted for the $\mathcal{ALC}$ language is based on Tarski-style semantics.

An interpretation $\mathcal{I}$ is a pair $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, where Δ is a non-empty set called the domain of the interpretation and $\cdot^{\mathcal{I}}$ is the interpretation function. The interpretation function maps:

- Each atomic concept A to a subset $A^{\mathcal{I}}$ of Δ.
- Each atomic role R to a subset $R^{\mathcal{I}}$ of $\Delta \times \Delta$.
- Each object name a to an element $a^{\mathcal{I}}$ of Δ.

The interpretation function can be inductively extended to concept descriptions as follows:

- $\bot^{\mathcal{I}} = \emptyset$
- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
- $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \exists b \in \Delta^{\mathcal{I}}, (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
- $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \forall b \in \Delta^{\mathcal{I}}, (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$

In DLs, the basic reasoning service on concept expressions is *subsumption*, written $C \sqsubseteq D$. This inference service checks whether the first concept always denotes a subset of the set denoted by the second one. We use this service

on the optimization of merged ontologies. Another service that we are using intensively is consistent checking of a knowledge base, i.e. an ABox A is consistent with respect to a TBox T, if there is an interpretation that is a model of both A and T [2].

Both domains, FCA and DL ontologies, use the term of *concept*. In the rest of this paper, concepts in the context of FCA (resp. DL ontology) are named formal concepts, resp. DL concepts. To clarify the distinction between them, we can state that DL concepts correspond to the attributes of $\mathcal{K}$.

1.2.3 Possibilistic logic

Possibilistic logic, or possibility theory, [12] provides an efficient solution for handling uncertain or prioritized formulas and coping with inconsistency. In this theory, each formula is associated to a real value in [0,1]. The notion of possibility distribution π is fundamental to define this logic's semantics and defined as $\pi : \Omega \rightarrow [0, 1]$, where Ω represents the set of all classical interpretations.

From a possibility distribution, two important measures can be processed: (i) the possibility degree of a formula ϕ, defined as $\Pi(\phi)= \max\{\pi(\omega) \mid \omega \models \phi\}$, where $\omega(\phi)$ is the degree of compatibility of interpretation ω with available beliefs. (ii) the certainty degree of a formula ϕ, defined as $N(\phi)$=1 - $\Pi(\neg\phi)$.

A possibilistic formula is a pair (ϕ, α) where ϕ is a logic formula and α expresses a degree of certainty. A set of possibilistic formulas, also called a possibilistic knowledge base (PKB), has the form $\{(\phi_i, \alpha_i)\}$ with $1 \leq i \leq n$ The classical knowledge base (CKB) associated with PKB corresponds to $\{\phi_i \mid (\phi_i, \alpha_i) \in PKB\}$. A PKB is consistent if and only if its CKB is consistent.

Given a PKB and $\alpha \in [0, 1]$, the α-cut of PKB is:
$PKB_{\geq\alpha} = \{ \phi \in CKB \mid (\phi, \beta) \in PKB$ and $\beta \geq \alpha\}$.

The inconsistency degree of PKB, denoted $Inc(PKB)$, is defined as $Inc(PKB) = \max\{\alpha_i \mid PKB_{\geq\alpha}$ is inconsistent$\}$.

Recently, possibilistic logic has been studied in the context of DL [13], [22], [27]. In Section 1.4, we exploit some of these results.

1.3 Ontology merging using FCA

In this section, we present the process of merging two source ontologies. In fact, the method can be used for several ontologies as long as these ontologies share elements of their datasets. That is ABoxes of these ontologies contain assertions about the same objects.

1.3.1 Source TBoxes

Let us consider two geographical applications that manipulate space parcel data. Each application uses an independent ontology formalism to represent the concepts related to its data. Also the teams of experts that designed each ontology may not agree on the semantics of some concepts.

Nevertheless, the two applications need to exchange information, and thus require that some correspondences are discovered between their DL concepts. The following two ontology extracts, O_1 and O_2, are used all along this paper. In order to ease the understanding and reading of our example, all concepts and roles are under scripted with the number of their respective ontology, i.e. '1' for O_1 and '2' for O_2.

Terminological axioms of ontology O_1:

(1) $CF_1 \equiv F_1 \sqcap \exists vegetation_1.C_1$
(2) $BLF_1 \equiv F_1 \sqcap \exists vegetation_1.M_1$
(3) $C_1 \sqcap M_1 \sqsubseteq \bot$

This extract of ontology O_1 defines two concepts, CF_1, standing for Coniferous Forest, and BLF_1, standing for Broad Leaved Forest, in terms of the concepts F_1 (Forest), C_1 (Coniferophyta) and M_1 (Magnoliophyta). Line #1 states that the coniferous forest concept is defined as the intersection of the concept Forest of O_1 and the concept having at least one vegetation being a coniferophyta. Line #2 defines the concept of a broad leaved forest accordingly with magnoliophyta. Line #3 states that the concepts coniferophyta and magnoliophyta are disjoint.

Terminological axioms of ontology O_2:

(4) $CF_2 \equiv F_2 \sqcap \forall vegetation_2.C_2 \sqcap \ \exists vegetation_2.C_2$
(5) $BLF_2 \equiv F_2 \sqcap \forall vegetation_2.M_2 \sqcap \ \exists vegetation_2.M_2$
(6) $MF_2 \equiv F_2 \sqcap \exists vegetation_2.C_2 \sqcap \ \exists vegetation_2.M_2$
(7) $C_2 \sqcap M_2 \sqsubseteq \bot$

The study of O_2 emphasizes that designers do not entirely agree on the semantics of forest related concepts of O_1. On line #4, the concept of a coniferous forest is defined has being a forest composed of at least coniferophyta vegetation and exclusively of this kind of vegetation. Line #5 defines the concept of broad leaved forest accordingly with magnoliophyta. In order to represent other kinds of forests, the designers of O_2 define a mixed forest concept as the intersection of being a forest with at least one coniferophyta vegetation and at least one magnoliophyta vegetation. Finally Line #8 states that the concepts coniferophyta and magnoliophyta of O_2 are disjoint.

Merging the ontologies O_1 and O_2 with some other ontologies would require that the TBoxes for these new ontologies are available and are no more expressive than $\mathcal{ALC}$.

1.3.2 Source ABoxes

Given the kind of TBoxes presented in the previous section, e.g. $\mathcal{ALC}$, we consider DL knowledge bases with non-empty ABoxes. In a first step, we map the information of the two ABoxes on a common set of observed objects.

The information of these ABoxes can be stored in a structured or unstructured format. It is interesting to note that the activity of several research teams in the DL and Semantic Web community focuses on studying cooperations between the domains of databases and knowledge bases represented in a DL. For instance, the authors of [26] recently claimed that the ideal solution would be to have the individuals of the ABox stored in a relational database and represent the schema of this database in a DL TBox. Also tackling this same objective, the team supporting the Pellet reasoner, one of the most popular OWL reasoners, recently released *OWLgres* which is being defined by their creators as a 'scalable reasoner for OWL2' (the latest version of the OWL). A main objective of this tool is to provide a conjunctive query answering service using SPARQL and the performance properties of relational database management systems. Hence, using such an approach, the set of observed objects may be retrieved from existing relational database instances.

The mapping we propose between both ontologies can be represented by a *matrix*, either generated by a specific tool and/or by interactions with end-users. In order to map concepts of both ontologies via the selected set of observed objects, a reference reconciliation tool may be used [10]. Using an approach that exploits a relational database as the data container for the ontology ABox enables to use existing FCA tools. This is the case of the ToscanaJ suite [4] which provides features for database connectivity.

We present a sample of this mapping in Table 1.1: the rows correspond to the objects of $\mathcal{K}$, i.e. common instances of the KB's ABox, and are identified by integer values from 1 to 6 in our example. In the context of geographical information, these values identify spatial parcels. The columns correspond to FCA attributes of $\mathcal{K}$, i.e. concept names of the two TBoxes. In the same table, we present, side by side, the formal concepts coming from our two ontologies, i.e. CF_1, BLF_1, F_1 from O_1, and CF_2, BLF_2, MF_2, F_2 from O_2. Thus this matrix characterizes the type of spatial parcels in terms of two different ontologies.

Merging more than two ontologies would require that the individuals of the ABox belong to the extension of the concepts of these ontologies. That is

concepts from a third ontology can be added to the columns of Table 1.1 and objects of the ABox (rows of the table) are instances of these new concepts.

1.3.3 Generation of the Galois connection lattice

The matrix is built using the information stored in the TBox and ABox of both ontologies:
- first, for each row, mark the columns where a specific instance is observed, e.g. the object on line #1 is an instance of the CF_1 and CF_2 concepts. Thus ABox information is used in this step.
- then, complete the row with the transitive closure of the subsumption relation between ontology concepts, e.g.: line #1 must be also marked for DL concepts F_1 and F_2, as respective ontologies entail that: $CF_1 \sqsubseteq F_1$ and $CF_2 \sqsubseteq F_2$. Here, the concept hierarchy of TBoxes is exploited.

Table 1.1 Sample dataset for our ontology merging example

	CF_1	BLF_1	F_1	CF_2	BLF_2	MF_2	F_2
1	x		x	x			x
2	x		x	x			x
3	x		x			x	x
4		x	x		x		x
5		x	x		x		x
6		x	x			x	x

It is interesting to note that lines #3 and #6 emphasize different assumptions for their respective parcels. For instance, the parcel corresponding to line #3 has been defined as a coniferous forest using the classification of O_1 while, possibly due to a vegetation not limited to coniferophyta, it has been defined as a mixed forest using O_2. The same kind of approach applies to the parcel associated to line #6.

Using Table 1.1 with the Galois connection method [9], we obtain the lattice of Figure 1.1, where a node contains two sets: a set of objects (identified by the integer values of the first column of our matrix) from $\mathcal{K}$ (extension), and a set of DL concepts from the source ontologies (intension), identified by the concept labels of source ontologies.

1.3.4 Dealing with emerging concepts

In order to concentrate solely on the intensional aspect of the lattice, i.e. the TBox, we now remove the extensional part of each node of the lattice.

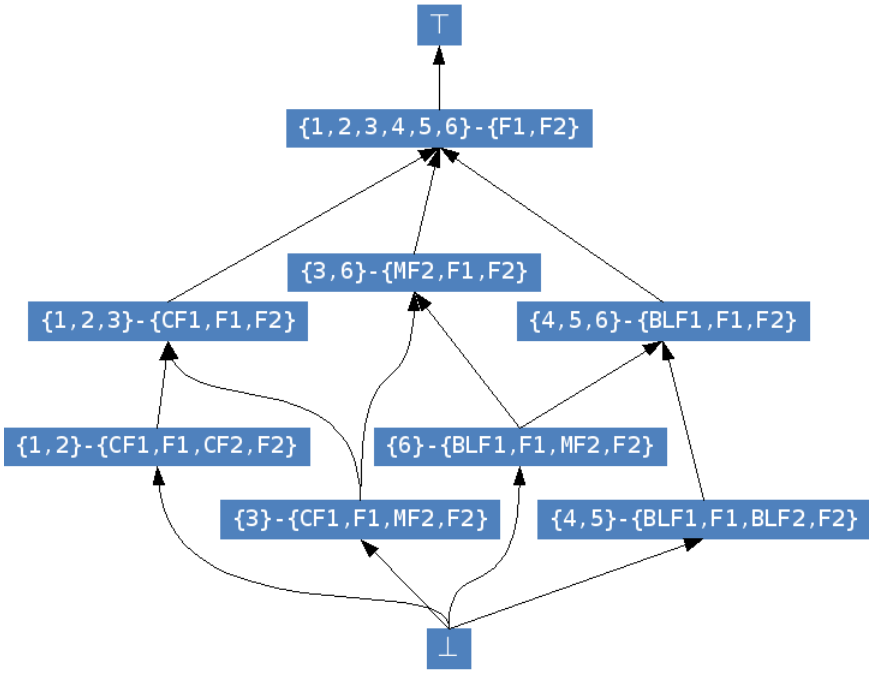

Fig. 1.1 Galois connection lattice

Hence, the only set present in each node correspond to concept names (Figure 1.2). Considering that the relationship holding between two nodes in this lattice corresponds to an inheritance property, it is possible to minimize each node's set by removing concept names that are present in an inherited node. The method we propose consists in deleting repeated occurrences of a given concept name along a path of the lattice and thus to obtain a minimal set of concept names for each node. Next, we define this notion of minimality:

Definition 3: Given a node N in the Galois connection lattice and a set of concept symbols S contained in its intension fragment. We consider that S is minimal for N if and only if there is no S' for N such that $|S'| < |S|$, where $|S|$ denotes the size of S.

Due to the lattice structure obtained by applying the Galois connection method, we can proceed by using a top-down navigation, i.e. starting from the top concept (Top), on the concepts of the merged ontology. Basically, this algorithm (named *optimizeLabel* and presented in Algorithm 1) proceeds as follows: for a given formal concept C of the lattice, it computes all its children c (line #1) and checks if a concept symbol used to characterize C is used in the concept name set for c (line #2). If this the case, this symbol is removed from their set of c (line #3) otherwise the set of symbols of c remain unchanged. Finally, the method is applied recursively to each concept c until

all concepts are processed (line #5).

Algorithm 1 optimizeLabel (Concept C)	
1	FOR EACH child c of C DO
2	IF label(C) $\in$ label(c) THEN
3	remove label(C) from label(c)
4	END IF
5	optimizeLabel(c)
6	END DO

Processing this algorithm on our running example, yields Figure 1.2 where lattice nodes contain singleton sets, corresponding to concept names from some of the source ontologies or newly introduced symbols, e.g. α, which replace empty sets. Several kinds of nodes, in terms of the size of a name set, can be generated with this method. Basically, it is important to distinguish between the following three kinds of nodes:

1. a singleton: a name of a concept from some of the source ontologies, because it can be distinguished from any of its successors by this specific name, e.g. this is the case for the $\{CF_1\}$. lattice node.
2. an empty set, denoted by a variable (α), because it can not be directly distinguished from any of its possible successors. We have 2 such nodes in Figure 1.2, namely α and β.
3. a set of several concept symbols, all belonging to source ontologies, because the mediation based on the given ABoxes, has not been able to split the concepts into several nodes. Indeed, it is as if the two names are glued together in a single concept name. In our running example, we have one such node with concept set $\{F_1, F_2\}$.

All singletons are maintained in the resulting merged ontology and we are now aiming to provide a concept description to the remaining concepts, case 2 and 3 of our node categorization. The first step toward our solution is to expand the concepts of the merged ontology according to their respective TBoxes [2]. That is, we replace each occurrence of a name on the right hand-side of a definition by the concepts that it stands for. A prerequisite of this approach is that we are dealing with acyclic TBoxes. Thus this process stops and the resulting descriptions contain only primitive concepts on the right hand-side.

We first deal with the nodes which are formed of several concept symbols, denoted σ_i, e.g. the node labeled F_1, F_2 in Figure 1.2. Due to the fact that the algorithm adopted results from the generation of the Galois connection lattice [9], these nodes appear at the top of the lattice and do not have multiple inheritance to concepts that are not of this form. Thus we adopt a top-down approach from the top concept ($\top$) of our merged ontology. We consider that the concepts associated are equivalent, e.g. $F_1 \equiv F_2$, since they have exactly the same extension. We also propose a single concept symbol σ, e.g. F (Forest) for F_1, F_2, and associate information to this concept stating

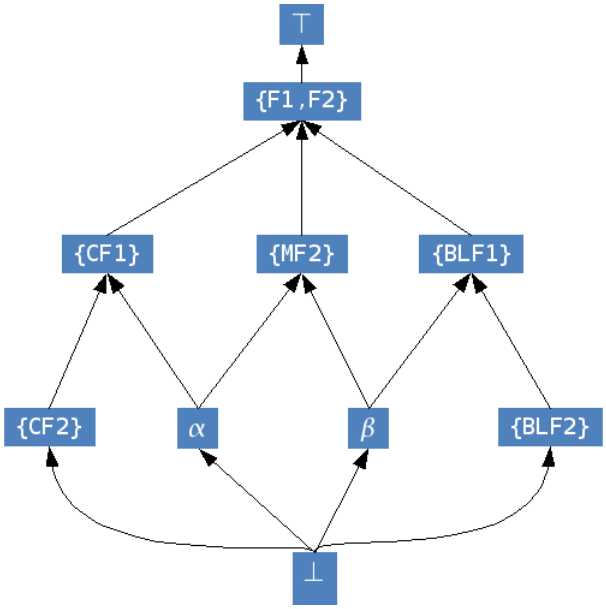

Fig. 1.2 Galois connection lattice with "empty nodes"

that this concept is equivalent to the original concepts for interoperability reasons, e.g. $F \approx F_1$ and $F \approx F_2$. Now all occurrences of the concept σ_i are translated into the concept symbol σ in the concept descriptions of the merged ontology.

We can now concentrate on the nodes with empty sets, e.g. α and β. According to the Galois based lattice creation, these nodes can not be at the root of the lattice. This means that they inherit from some other concept(s). We use the description of these inherited concept(s) to provide a description. Using this method, the concepts α and β of Figure 1.2 have the following description:
$\alpha \equiv CF_1 \sqcap MF_2 \equiv F \sqcap \exists vegetation_1.C_1 \sqcap \exists vegetation_2.C_2 \sqcap \exists vegetation_2.M_2$
$\beta \equiv BLF_1 \sqcap MF_2 \equiv F \sqcap \exists vegetation_1.M_1 \sqcap \exists vegetation_2.C_2 \sqcap \exists vegetation_2.M_2$

All concepts from the merged ontology have been associated to a concept description, except of course the primitive concepts. Alignments between primitive concepts and roles of the source ontologies are able to refine the merged ontology. Later in this section, we will propose solutions to finding these alignments and dealing with their uncertainty, but we now present the impact of providing such correspondences between TBox elements.

Suppose that we are being provided the following alignments: $C_1 \equiv C_2$, $M_1 \equiv M_2$ and even $vegetation_1 \equiv vegetation_2$. So we can easily introduce some concept symbols to simplify the different equivalences:

(8) $C \equiv C_1 \equiv C_2$, $M \equiv M_1 \equiv M_2$ and $vegetation \equiv vegetation_1 \equiv vegetation_2$

We are then able to modify the descriptions of the merged ontology and we denote this TBox as O_{m1}:

(9) $CF_1 \equiv F \sqcap \exists vegetation.C$
(10) $BLF_1 \equiv F \sqcap \exists vegetation.M$
(11) $CF_2 \equiv CF_1 \sqcap \forall vegetation.C \sqcap \ \exists vegetation.C$
(12) $BLF_2 \equiv BLF_1 \sqcap \forall vegetation.M \sqcap \ \exists vegetation.M$
(13) $MF_2 \equiv F \sqcap \exists vegetation.C \sqcap \ \exists vegetation.M$
(14) $\alpha \equiv F \sqcap \exists vegetation.C \sqcap \ \exists vegetation.M$
(15) $\beta \equiv F \sqcap \exists vegetation.C \sqcap \ \exists vegetation.M$
(16) $C \sqcap M \sqsubseteq \bot$

We can notice that the descriptions for the concepts α, β and MF_2 are the same. Thus we can state that $MF_2 \equiv \alpha \equiv \beta$. Finding such equivalences, or subsumption relationships, is easily processed by a DL reasoner. This result is comforted by the fact that starting from the ontologies O_1 and O_2 and the alignments of (8), any DL reasoner is able to provide the ontology O_{m1}, assuming that we have the alignment $F \equiv F_1 \equiv F_2$ (which has been deduced from our Galois lattice). The lattice corresponding to this new ontology is depicted in Figure 1.3.

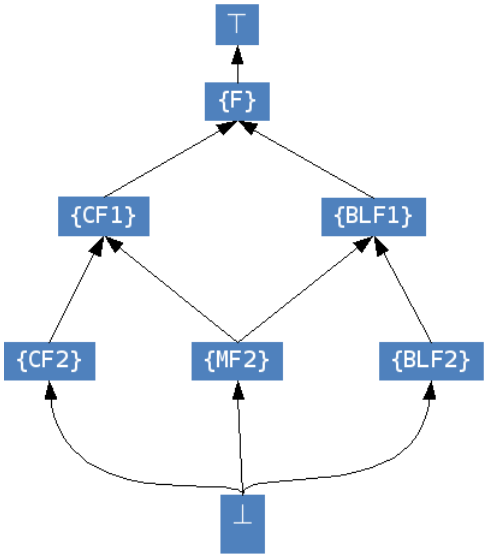

Fig. 1.3 Lattice corresponding to merged ontology O_{m1}

Of course, alignments different from (8) can be proposed between primitive concepts and roles of O_1 and O_2. For instance, if we consider the alignments

in (17), then the optimized merged ontology again correspond to Figure 1.3.
(17) $M_2 \sqsubseteq M_1, C \equiv C_1 \equiv C_2$ and $vegetation \equiv vegetation_1 \equiv vegetation_2$

Concentrating on the relationships between M_1 and M_2, alignments other than (8) and (17) can generate different merged ontologies. Let consider the alignments in (18) where the only difference with (8) and (17) is that now M_1 is a subconcept of M_2:
(18) $M_1 \sqsubseteq M_2, C \equiv C_1 \equiv C_2$ and $vegetation \equiv vegetation_1 \equiv vegetation_2$

Then looking at the descriptions of BLF_1 and BLF_2 (respectively (2) and (4) in Section 1.3), we can no longer state that $BLF_1 \sqsubseteq BLF_2$. We consider that the alignments of (18) do not contradict our FCA-based method but instead refine the constructed lattice of Figure 1.2. In fact, this lattice is the result of applying a Galois connection based algorithm from a given dataset. This dataset can be considered to be a model of the merged ontology but it is only one of the possible models for this ontology. The statements in (18) say that instances of BLF_2 need not to be instances of BLF_1, a situation that was not present on our dataset (Table 1.1).

Moreover, the statements of (18) allow us to state that $MF_2 \sqsubseteq CF_1$ and $\alpha \equiv MF_2$ but prevent us from saying that $MF_2 \sqsubseteq BLF_1$. The lattice corresponding to this new merged ontology, which we denote O_{m2}, is presented in Figure 1.4.

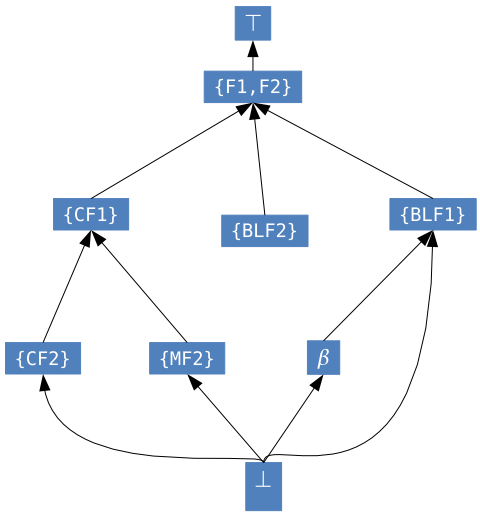

Fig. 1.4 Lattice corresponding to merged ontology O_{m2}

In terms of the DL model-theoretic semantics presented in 1.2.2, the $MF_2 \sqsubseteq \neg BLF_1$ axiom makes the ABox represented in Table 1.1 inconsistent with the merged ontology of Figure 1.4. Recall that an ABox $\mathcal{A}$ is consistent

with respect to a TBox $\mathcal{T}$, if there is an interpretation that is a model of both $\mathcal{A}$ and $\mathcal{T}$. Intuitively, $MF_2 \sqsubseteq \neg BLF_1$ states that it is not possible to be an instance of both BLF_1 and MF_2 in a given model which is not case of object #6 in our dataset.

This raises the issue of the confidence one has on the existence of an object, of an alignment and to their relationship. For instance, we can have a greater confidence on the statements of (18) than on the existence of object #6. This would yield a merged ontology similar to the one presented in Figure 1.4 but without the β concept. We will provide details on the notion of confidence values when introducing our solution to deal with uncertainties in Section 1.4.

In summary, we can generate different ontologies based on the fact that we are able to propose different alignments, to assign them confidence values and to assign confidence values to some objects of our sample dataset matrix. In order to provide alignments between source ontologies, we consider the following two approaches: these alignments originate from external ontologies or they are provided by the end-user.

1.3.4.1 Alignments originating from external knowledge

The alignments of primitive concepts and roles can be provided by an external knowledge source. This is in fact frequently the case when designing ontologies. Early in the design process, a background ontology, preferably recognized as a standard in the application domain, is identified and imported in the source ontologies. It is likely that the source ontologies we consider for fusion, import some common parts of a given background ontology, e.g. it can be the case in spatial information with the SWEET ontology. Then the alignment of some imported primitive concepts and roles is straightforward and less subject to some uncertainty since their interpretations are identical.

1.3.4.2 End-user defined alignments

In cases alignments can not be provided by some background knowledge, end-users can define their own correspondences between concepts and roles. In such a situation, different end-users may provide differing alignments. Also, an end-user may not be totally confident on an alignment she is providing. This uncertainty aspect needs to handled by the system in order to propose the most adequate merged ontology.

1.4 Dealing with uncertainty

In the previous section, we highlighted several situations characterized by some forms of uncertainty. In particular, we highlighted uncertainties at the 'object-level', that is we are not totally confident in the correctness of some of our dataset objects. We also emphasized on uncertainties at the 'alignment-level', that is one can be more or less confident on the correspondences set between concepts and roles of the source ontologies. In order to deal with these uncertainties, we use possibilistic logic to encode both object and alignment confidences within a DL knowledge base context.

Concerning the setting of confidences on objects of the source datasets, we do not believe that an automatic solution can produce reliable and relevant confidence values. Hence, it is necessary to integrate the end-user, generally a domain expert, in the process of setting these certainty levels. Two solutions can be envisioned: (i) ask the end-user to assign confidence values to all the tuples of the dataset, (ii) assume that the dataset is sound and ask the end-user to set certainty degrees only on the tuples that are causing inconsistencies.

Solution (i) can not be realistically implemented since the dataset may be very large and the end-user may not have the time and knowledge to assign a confidence value to each tuple. In this perspective, solution (ii) is much more realistic and efficient since we are asking the end-user to study only a subset of the dataset objects. This is based on the assumption that the data contained in practical databases is sound and that only a subset of it is erroneous. Hence, this approach requires that all objects are first set to a default value of 1, i.e. assuming soundness, recall that confidence are set in [0,1]. It also implies that the system provides a solution to check consistency of the knowledge base and is able to identify objects responsible for inconsistencies. Such a solution is already implemented in several DL reasoners, e.g. Pellet. Once the knowledge base has been detected as inconsistent, we invite the end-user to refine the confidence value of each object responsible for the knowledge base inconsistency.

The next question to ask ourselves is: when to check the consistency of the (merged) knowledge base ? In fact, this knowledge base can only be detected inconsistent after the application of some alignments. This is due to the consistency of the merged ontology computed from our Galois connection based solution.

We will come back to this inconsistency aspect but first, we would like to make precise the definition of uncertainties on the alignments. We consider that alignments originating from some external knowledge or deduced by our FCA solution (e.g. $F_1 \equiv F_2$) are set with a default value of 1. This assumption is motivated by the following facts:

- the quality of the external ontology generally imported in specific ontologies. That is, we consider the import of an ontology fragment as a strong

end-user commitment which ensures the adequacy and quality of this external ontology.
- in practice, our FCA solution only computes concept equivalence on large concept extensions which are likely to be correct.

Nevertheless, the end-user has the ability to refine confidence values on any alignment. Each alignment proposed by the end-user requires a confidence value which can only be defined manually.

Consider our running example and the alignments of (18), we can define the following set of possibilistic formulas for our alignments :
$\{(F_1 \equiv F2,\ 1),(M1 \sqsubseteq M_2,\ 0.5),(C_1 \equiv C_2,\ 0.9),(vegetation_1 \equiv vegetation_2,\ 1)\}$. That is, we are totally confident on the following alignements: $F_1 \equiv F_2$ and $vegetation1 \equiv vegetation_2$. But to certain extent, we are not totally confident of the correctness of $C_1 \equiv C_2$ and $M_1 \sqsubseteq M_2$ since their degree of certainty are respectively of 0.9 and 0.5.

The process of generating a consistent merged ontology with respect to a set of alignments and some certainty levels can be defined by the following algorithm.

Algorithm 2	createOntology (Ontology O, Alignment Al, Dataset D)
1	create an ontology O' from O and Al.
2	create an ABox A' from O' objects of D
3	WHILE ($< O', A' >$ is inconsistent)
4	$I(D)$ = inconsistent set of objects of $< O', A' >$
5	Ask end-user to set confidence values to entries of $I(D)$
6	END WHILE
7	classify O'
8	return O'

The understanding of this algorithm is relatively straightforward. Our FCA-based solution generates a merged ontology which is later refined by a set of alignments (step 1). Moreover, the object matrix of our source ontologies is transformed into an ABox (step 2). All axioms are associated with a certainty degree which makes this knowledge of a possibilistic one. Initially, these certainty levels correspond to the value 1 for all concepts and objects of the knowledge base and axioms of the alignments are defined by the end-user. We now need to clarify the notion of consistency checking of step 3 in the context a possibility logic theory. The notion of consistency of a possibilistic knowledge base (PKB) is related to its possibility distribution, denoted π_{PKB}, see Section 1.2.

Adapted to the DL context, a PKB corresponds to $\langle PTBox, PABox\rangle$ where $PTBox$ and $PABox$ are respectively a possibilistic TBox and ABox. The classical DL axioms associated to $PTBox$ (resp. $PABox$) is $TBox$, i.e. $\{\phi_i \mid (\phi_i, \alpha_i) \in PTBox\}$ (resp. $ABox$ defined similarly) and $KB = \langle TBox, ABox\rangle$. With $\alpha \in [0, 1]$, the α-cut of $PTBox$ is (defined similarly for $PABox$):

$PTBox_{\geq\alpha} = \{\ \phi \in TBox \mid (\phi, \beta) \in PTBox$ and $\beta \geq \alpha\}$. Thus, $PKB_{\geq\alpha} = \langle PTBox_{\geq\alpha}, PABox_{\geq\alpha}\rangle$.

The possibility distribution of an interpretation I of PKB can be defined as follows:

$$\pi_I = \begin{cases} 1 & \text{if } \forall\phi_i \in PKB, I \models \phi_i \\ 1 - max\{\alpha_i | I \not\models \phi_i, (\phi_i, \alpha_i) \in PKB\} & \text{otherwise} \end{cases}$$

Then a PKB is consistent if and only if $\pi_{PKB} \models$ PKB and we are now able to compute the consistency checking the step 3 of our algorithm.

In order to identify the instances responsible for the inconsistencies, we use the *instance checking* inference service which states that an individual a is a plausible instance of a concept C wrt a PKB if $PKB_{>Inc(PKB)} \models C(a)$, where $PKB_{>\alpha}$, the strict α-cut, is defined as follows: $\{\phi \in CKB | (\phi, \beta) \in PKB$ and $\beta > \alpha\}$.

In the context of our running example with alignments (18), the first two lines of the *createOntology* algorithm generates the O_{m2} ontology with an ABox containing the 6 objects of the Table 1.1. This classical knowledge base (CKB) is inconsistent since the intersection of MF_2 and BLF_1. In fact a standard DL reasoner is able to identify object #6 (denoted obj#6) as a source of this inconsistency. In step (5) of our our algorithm, the end-user is proposed to modify the certainty level associated to object #6. Suppose that the end-user is aware that some parcels have been erroneously classified or that some sensors were not really accurate during some field experiments and sets the certainty level of this object to a value of 0.3. We now concentrate on two concepts which have object #6 in their extensions: MF_2 and β. The possibility value of MF_2 (resp. β) is the maximum possibility value of objects #3 and #6, i.e. max{1,0.3}=1, (resp. maximum possibility value of object #6, i.e. max{0.3}=0.3).

Let α=0.3, we have $PKB_{\geq 0.3} = \{PTBox_{\geq 0.3}, PABox_{\geq 0.3}\}$ where the formulas $MF_2 \sqsubseteq \neg BLF_1$, $\beta \sqsubseteq BLF_1$ are contained in $PTBox_{\geq 0.3}$ and $PABox_{\geq 0.3}$ contains the assertions $MF_2(obj\#6)$ and $\beta(obj\#6)$ (respectively stating that object #6 is an instance of MF_2 and β). It is clear that $PKB_{\geq 0.3}$ is inconsistent. Now let α=0.5, then $PKB_{\geq 0.5} = \{PTBox_{\geq 0.5}, PABox_{\geq 0.5}\}$ where $PTBox_{\geq 0.5}$ contains the formula $MF_2 \sqsubseteq BLF_1$ and $PABox_{\geq 0.5}$ contains the assertions $MF_2(obj\#6)$ $PKB_{\geq 0.5}$ is clearly consistent. Therefore Inc(PKB)=0.3. Hence, our method enables to compute several merged ontology based on a set of given alignments and interactions with end-users to specify possibility values of certain objects.

1.5 Related work

In this Section, we survey related works in ontology mediation solutions and in particular we present some solutions which exploit extensions of the ontologies, i.e. ABoxes.

In the literature, two distinct approaches in ontology merging have been distinguished. In the first approach, the merged ontology captures all the knowledge of the source ontologies and replaces them. An example of such a system is presented in [25] with the PROMPT tool. In the second approach the source ontologies are not replaced by the merged ontology, but rather a so-called 'bridge ontology' is created. The bridge ontology imports the original ontologies and defines the correspondences using axioms which are called "bridge axioms". An example of such an approach is the Ontomerge solution which has been described in [11].

The most relevant work related to our solution is the FCA-merge system [28]. It uses instances of ontology classes to exploit an FCA algorithm. The FCA-merge system produces a lattice of concepts which relates concepts from the source ontologies. This new concept lattice is then handed to the domain expert in order to generate the merged ontology. Thus we can consider FCA-merge to be a semi-automatic solution while our solution aims to generate the merged ontology automatically. So the main differences are that the FCA-merge is unable to propose concepts emerging from the fusion of the source ontologies and does not propose a label generation solution. Also, without the help of domain experts, the FCA-merge system is not able to refine the merged ontology.

The Ontex (Ontology Exploration) method, presented in [19], also tackles the tasks of creating and merging ontologies using the knowledge acquisition technique of Attribute Exploration [18] encountered in FCA. For both ontology creation and merging, the Ontex method concentrates on providing a high quality conceptual hierarchy of the top-level concepts. Considering the merging task, Ontex provides an interactive knowledge acquisition technique for the top-level concepts. The other concepts of the of the merged ontology can be created using heuristics-based approaches. Comparatively, our approach does not limit the concept processes in terms of levels in the hierarchy.

Considering works involving FCA methods and DLs, it is interesting to study [3]. In this paper the authors are concerned with the completeness quality dimension of TBoxes, i.e. they propose techniques to enable ontology engineers in checking if all the relevant concepts of an application domain are present in a TBox. Like our approach, one of their concerns is to minimize interactions with domain experts. Hence FCA techniques are being used to withdraw trivial questions that may be asked to experts in case of incomplete TBoxes. The approach we presented in this paper is more concerned with the generation and optimization of a mediated ontology. We can also consider

that our approach is more involved in the soundness quality dimension and tackles the issue of generating different forms of merged ontology.

1.6 Conclusion

In this paper, we presented an approach to merge DL ontologies based on the methods of FCA. Our main contribution enables the creation of concepts not originally in the source ontologies and the description of the concepts in terms of elements of source ontologies. Moreover, through the management of several forms of uncertainty (at the object and alignment levels), with DL extended to possibility theory, we are able to easily handle the creation of different merged ontologies.

We have presented this approach in the geographical domain but it can be exploited in all fields where ontologies are used. We are currently testing its usefulness in the context of life science applications, i.e. medicine and pharmacology.

Future work on this system are related to extracting automatically an optimized set of instances from ABoxes for the Galois connection matrix. In particular, we would like to provide a notion of weights to objects of the matrix, i.e. the number of tuples from the source ABox that satisfy a given distribution over the source ontology concepts. For instance, the weight of object #1 in Table 1.1 would be 2 and we could remove object #2 from the matrix. This approach would enable to load more compact matrices in our solution. Moreover, we would like to pursue investigations in using possibilistic theory in the context of ontology mediation.

Another direction for future work consists in studying more expressive DLs, i.e. going beyond the $\mathcal{ALC}$ language. For instance, we aim to study the $\mathcal{SHIF}$ and $\mathcal{SHOIN}$ DLs which underpin the *Lite* and *DL* OWL species. This would open our solution to an important number of existing and widely used ontologies designed for the Web.

References

1. AdV (1998). Amtliches Topographisch Kartographisches Informatoinssyetm ATKIS. Technical report, Landdesvermessungsamt NRW, Bonn.
2. F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*, New York, USA: Cambridge University Press, 2003.
3. F. Baader, B. Ganter, B. Sertkaya, U. Sattler, "Completing Description Logic Knowledge Bases Using Formal Concept Analysis" in *Proc.IJCAI'07* 2007, pp 230-235
4. P. Becker, J.-H. Correia, "The toscanaj suite for implementing conceptual information systems" in *Formal Concept Analysis State of the Art, Berlin Heidelberg.* Springer.

5. G. Birkhoff, *Lattice Theory*. American Mathematical Society, Colloquium Publications, 3rd edition, 1973.
6. V. Choi, "Faster Algorithms for Constructing a Concept (Galois) Lattice" , CoRR (abs/cs/0602069) 2006
7. B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, U. Sattler, "OWL 2: The next step for OWL". J. Web Semantics, 6 (4) 2008. pp309-322.
8. O. Curé, R. Jeansoulin, "An FCA-based Solution for Ontology Mediation" in *Proc.ONISW'08*, pp. 39-46.
9. B. Davey, H. Priestley, *Introduction to lattices and Order* New York, Cambridge University Press, 2002.
10. X. Dong, A. Halevy, J. Madhavan, "Reference reconciliation in complex information spaces", in *Proc.SIGMOD '05* 2005. pp 85-96.
11. D.Dou, D. McDermott, P. Qi, "Ontology translation by ontology merging and automated reasoning" in *Proc.EKAW'02* 2002,pp3-18.
12. D. Dubois, J. Lang, H. Prade, "Possibilistic logic". in *Handbook of logic in artificial intelligence and logic programming*, Oxford University Press, 2004. pp 439-513.
13. D. Dubois, J. Mengin, H. Prade, "Possibilistic uncertainty and fuzzy features in description logics. A preliminary discussion" in *Capturing Intelligence: Fuzzy Logic and the Semantic Web* pp101-113. Elsevier 2006.
14. M. Ehrig, *Ontology Alignment: Bridging the Semantic Gap*, Springer-Verlag, Heidelberg (DE), 2006.
15. Corine land cover, technical guide. Technical report, European Environmental Agency. ETC/LC, European Topic Centre on Land Cover.
16. J. Euzenat, P. Shvaiko, *Ontology matching*, Springer-Verlag, Heidelberg (DE), 2007.
17. B. Ganter, R. Wille, *Formal Concept Analysis: mathematical foundations* New York, USA: Springer-Verlag, 1999
18. B. Ganter, "Attribute Exploration with Background Knowledge". TCS 217(2), 1999, pp 215-233.
19. B. Ganter, G. Stumme, "Creation and Merging of Ontology Top-Levels". in *Proc.ICCS '03* 2003 pp 131-145
20. Y.Kalfoglou, M.Schorlemmer, *Ontology mapping: the state of the art*. Knowledge Engineering Review, 18(1), 2003, pp1-31.
21. P.C. Kanellakis. Elements of relational database theory *Handbook of theoretical computer science (vol. B): formal models and semantics*:1073–1156, MIT Press, 1990.
22. T. Lukasiewicz, U. Straccia, "Managing uncertainty and vagueness in description logics for the Semantic Web", Journal Web Semantics, 6 (4), 2008 pp291-308.
23. D. McGuinness. "Ontologies Come of Age". In D. Fensel, J. Hendler, H. Lieberman, W. Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. Cambridge, USA. MIT Press, 2003.
24. B. Motik, I. Horrocks, U. Sattler, "Bridging the gap between OWL and relational databases" in *Proc.WWW'07* 2007.
25. N. Noy, M. Musen, "PROMPT: Algorithm and tool for automated ontology merging and alignment" in *Proc.AAAI'00*, 2000.
26. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, *Linking Data to Ontologies*. Journal of Data Semantics, 10, 2008. pp 133-173.
27. G. Qi, J. Pan, Q. Ji, "A Possibilistic Extension of Description Logics", in *Proc.DL'07* 2007
28. G. Stumme, A. Maedche, "FCA-MERGE: Bottom-Up Merging of Ontologies" in *Proc.IJCAI'01* 2001, pp225-234.
29. Semantic Web for Earth and Environmental Terminology (SWEET). http://sweet.jpl.nasa.gov/ontology/